LSTM Model for Drift-Resilient Time Series Forecasting.

Wai Lee Boo 2625170
Supervised by: Professor Leandro Minku

1 Background

Forecasting financial time series is a long-standing challenge due to the inherently non-stationary and volatile nature of market data. Traditional and even deep learning models such as ARIMA and basic LSTM neural networks often fail to maintain accuracy when the underlying data distribution changes (also known as concept drift). As a result, the need for a model that is resilient to drift and adaptive to the evolving market has become one of the central focus in financial market forecasting. [9]

Recently, researchers have explored the use of dynamic swarm intelligence and adaptive ensemble learning to address this issue. Oliveira et al. [8] proposed a Dynamic Swarm Intelligence framework based on Extreme Learning Machines (ELMs) (a single hidden layer feed-forward neural network) that enables models to detect, adapt to, and reuse prior knowledge when drift occurs. by leveraging the collective behaviour of multiple particles. Similarity, Azeem et al. [2] developed an Adaptive Ensemble LSTM (AE-LSTM) for load forecasting on smart grids. Each LSTM instance operates with distinct initialisations. These framework continuously updates "trust weights" via ADWIN and PH drift detectors, allowing rapid adaptation without full retraining. However, such approaches may struggle to capture entirely new patterns if the data distribution shifts drastically.

In the financial domain, deep learning architectures have been widely used and have been shown to outperform traditional models. Han and Fu [5] demonstrated that Bi-Directional LSTM networks outperform the standard unidirectional model by capturing richer temporal dependencies and achieve higher accuracy in short-term market forecasting compared to unidirectional LSTM. In addition, recent studies

[1, 3, 4] highlight the influence of the tone and sentiment of the media on investor perception and market pricing. Model such as FinBERT, a domain specific transformer fine-tuned on financial text, provide highly accurate sentiment analysis and enhance the modelling of information driven market dynamics. However, sentiment alone cannot fully explain price movements, which indicates the need for hybrid architectures that integrate numerical and textual data for more comprehensive forecasting.

Similarly, Kumar et al. [7] introduced an Adaptive PSO–LSTM framework, in which a Particle Swarm Optimisation (PSO) algorithm is used to optimise the initial weights and biases of the LSTM gates and the output layer before training. This approach aims to mitigate poor weight initialisation, a common cause of local minima and slow convergence in gradient-based training, by providing globally optimised starting parameters before fine-tuning with Adam. Although this method improves convergence speed and final prediction accuracy, it remains an offline optimisation technique, trained on static datasets without explicit drift detection or online adaptation. In contrast, Davidovic et al. [6] integrated an Autoencoder-LSTM with the ADWIN drift detector, demonstrating that ADWIN can effectively identify distributional changes and trigger timely retraining, resulting in lower post-drift errors compared to the traditional recurrent model.

2 Problem Definitions

Although previous work has explored swarm intelligence, ensemble learning, and adaptive neural frameworks to improve robustness, most remain offline or architecture level optimisations without the ability to adapt dynamically once deployed. As a result, these models fail to maintain accuracy when exposed to continuously evolving market conditions.

Hence, the core problem addressed in the research is the lack of an adaptive, drift-resilient LSTM-based forecasting model capable of updating its parameters in real time when concept drift is detected while maintaining prediction accuracy and stability without requiring full retraining.

3 Aim

The aim of this project is to develop an LSTM-based time series forecasting model enhanced with Particle Swarm Intelligence (PSO) and news sentiment analysis to effectively adapt to concept drift in dynamic financial data. The model aims to remain accurate and stable by combining LSTM for time-series learning, PSO for optimisation, and news sentiment as an external signal, so it can adapt when market conditions change.

4 Project Description

4.1 Objective

The primary objective of this research is to design a drift-resistant forecasting framework that maintains high predictive accuracy despite continuous or abrupt changes in the data distribution. To achieve this, the study pursues the following specific objectives:

1. Develop and train an LSTM-based model to forecast the market value next day using historical financial data.
2. Incorporate sentiment features extracted from financial news (e.g. FinBERT) to enhance forecasting accuracy.
3. Baseline comparison.
4. Evaluation in increasing forecast horizons.
5. Conduct an ablation study to access the impact of each component (e.g. PSO optimisation, drift detection and sentiment integration on overall performance)

4.2 Possible Drift Detection used

The following list below shows different types of drift detector that can be implemented to detect concept drift. All this mainly uses the accuracy of the forecast (evaluation metrics in Section 5.5). However, KSWIN work slightly differently, as it looks directly at the input features (e.g. prices) to see if their behaviour has changed over time by keeping a sliding window of recent observations from the data stream, which in theory can warn sooner that the market is shifting.

1. ECDD
2. ADWIN
3. PHT
4. HDDM
5. KSWIN

4.3 Possible use of PSO(which to use?)

1. **Weight and Bias Initialisation:** PSO can be used to find optimal initial weights and biases for the LSTM model before gradient-based training. This helps the model converge faster and reduce the chance of getting stuck in local minima.
2. **Output Layer Re-optimisation:** During deployment, PSO can re-optimise the output layer parameter when a drift detector (Section 4.2) signal a change in data distribution. This allow the model to adapt to market condition without full retraining.

3. **Hyperparameters Tuning:** PSO can be used before the actual training (e.g., number of neurones, learning rate, batch size, how many past days to consider) to fine a well tuned LSTM configuration automatically.

4. **Hybrid Adaptation:** Small drift then use PSO (fast adaption example 2). If big drift consider deeper retraining to adjust the learned temporal patterns, backpropagation)

5 Methodology

5.1 Data & Preprocessing

Dataset Uses

1. **Market Data:** value like Open, Close, High, Low, Volume, Returns (S&P500, AAPL, MSFT, AMZN)

2. **Self made synthetic dataset:** Self generate dataset to simulate various concept drift scenarios to see how it perform

3. **FinBERT Compatible Dataset:** Consider convert classification into numerical sentiment features, which then merge with market data

5.2 Baseline Models

How many baseline model should I need? How many is too much?

1. LSTM

2. Bi-Directional LSTM

3. XGBoost

4. AE-LSTM

5. ADAM-LSTM

6. PSO-LSTM

7. *Others to be considered during development

5.3 Proposed Model

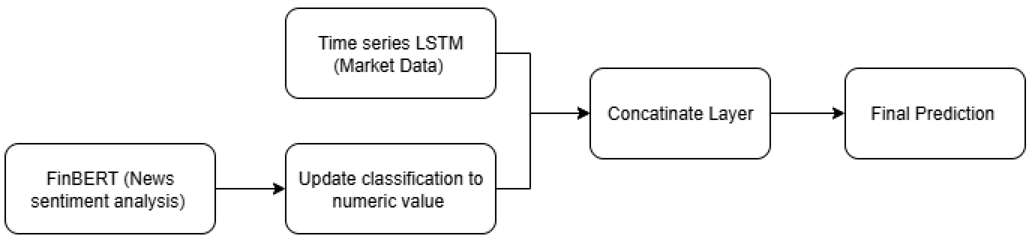

Figure 1: Model Architecture

5.4 Output

The model initially predicts a single target value, such as a high price, based on historical market data and sentiment input later on.

If this setup performs effectively, it can be extended to a multi-output configuration that is capable of forecasting the complete OHLV (Open, High, Low, and Volume) values. (**Optional:** Additionally, the framework can be adapted for multi-horizon forecasting, enabling prediction for longer periods such as the next week or subsequent trading intervals.)

5.5 Evaluation Metrics

The evaluation metrics below are used to assess the accuracy of the forecast and the overall performance of the model. These measures quantify how closely the predicted values align with actual outcomes and provide insight into the model's overall reliability and effectiveness. The same equation as in the following can also be used to predict multiple values. Additionally, these metrics can be used to test on synthetic datasets also.

1. Mean Absolute Error (MAE)
2. Root Mean Squared Error (RMSE)
3. Mean Absolute Percentage Error (MAPE)

Last Step: Consider including financial performance metrics (e.g. risk-adjusted measures) in the final evaluation state of the model. At this stage, a backtesting approach can be applied to simulate how the model would have executed trades based on its predictions. The simulated trading outcomes produce a series of returns that can subsequently be used to compute financial metrics using the formulas below:

1. Sharpe Ratio: risk adjusted return

Step to back-test

1. Start with a fixed starting capital (e.g., £10,000).
2. If the model predicts an increase, take a long position (buy). Otherwise, take a short position (sell).
3. Close the position at the next day's closing price.
4. Calculate the daily return:

$$r_t = \begin{cases} \dfrac{P_{t+1} - P_t}{P_t}, & \text{if long} \\ \dfrac{P_t - P_{t+1}}{P_t}, & \text{if short} \\ 0, & \text{if no trade} \end{cases}$$

5. Produce a sequence of returns based on the trading strategy:

$$r_1, r_2, r_3, \dots, r_N.$$

6. Calculate the average daily return:

$$\mu = \frac{1}{N}\sum_{t=1}^{N} r_t.$$

7. Calculate the daily volatility (standard deviation):

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}(r_t - \mu)^2}.$$

8. Compute the Sharpe Ratio (risk-adjusted return):

$$\text{Sharpe Ratio} = \frac{\mu - r_f}{\sigma},$$

where r_f is the risk-free rate (set to 0 for simplicity).

9. For the annualised Sharpe Ratio, multiply by $\sqrt{252}$:

$$\text{Sharpe}_{\text{annual}} = \frac{\mu - r_f}{\sigma} \times \sqrt{252}.$$

6 Task Breakdown

1. **Data Preparation (without news)**
   (a) Collect historical financial data (e.g. daily stock prices)
   (b) Pre-process:
      - Clean missing values (usually should not have any missing value)
      - generate sliding windows
      - Split data chronologically into train/validation/test

2. **Build some baseline LSTM**
   (a) These models will not include any drift-handling or sentiment integration mechanisms
   (b) Use their performance as reference benchmarks to evaluate later adaptive and hybrid models.

3. **PSO-LSTM Implementation (No Drift Detection)**
   - Select one PSO variant **Section 4.2**
   - Compare convergence speed and forecast accuracy with baseline LSTM (Can also consider as baseline)

4. **Drift detection and Dynamic Swarm integration**
    - Implement one or more concept drift detectors
    - Test detectors individually on synthetic drifting datasets to evaluate responsivity and false alarm rates
    - Integrate the best-performing detector into the PSO-LSTM pipeline
    - Evaluate the overall model after integrating
5. **Sentiment Integration Phase**
    (a) Collect and preprocess financial news data for the same assets and period as market data
    (b) Apply FinBERT or similar financial domain model to extract sentiment embeddings (positive, neutral, negative)
    (c) Consider converting sentiment output into numeric feature(might improve accuracy)
    (d) Assess whether sentiment inclusion improves prediction accuracy under drift and shock conditions
6. **Evaluation and Analysis**
    (a) Evaluate all models using metrics in **Section 5.5**
    (b) Test performance in synthetic and real-world drift scenarios
    (c) Analyse adaptation time, recovery speed, and post-drift stability for each approach
7. **Ablation Studies**
    (a) Conduct ablation experiments to isolate the effect of PSO, drift detection, and sentiment integration
8. **Documentation and Reporting**
    (a) Compile all results into a structured report
    (b) Discuss limitations and potential extensions

7 Risks & Mitigations

1. **High computational cost:** Running multiple models or swarm particles concurrently requires substantial processing power and memory, especially during retraining or optimisation.
    - Consider model pruning, early stopping, and reduced particle counts
2. **Degraded performance from noise or irrelevant news input:** Inaccurate or misleading news articles can distort sentiment signals and negatively impact model predictions.

- Implement robust news filtering using keywords and remove duplicate news

3. **Unnecessary retraining due to transient shocks:** Sudden short-term market fluctuations may falsely trigger retraining, leading to wasted computational effort and instability.
   - Consider introducing a confirmation window where drift must persist for several time before triggering retraining
   - Define minimum impact thresholds to ignore small deviations
4. **Over-fitting risks:** The model may over-adapt to particular drift events or time periods, limiting generalisation to unseen conditions.
   - Use early stopping based out-of-sample validation loss
5. **Drift Detector Instability:** Incorrect Tuning of drift detectors may lead to false positives or missed drift events.
   - Consider using a multiple drift detector to reduce false positives
6. **Hard to verify drift detection in real-time:** It can be difficult to determine when the concept drift mechanism has truly detected a distributional change, as ground truth drift points are unknown in live data.
   - Compared with and without drift detector model, if the adaptive model error is smaller, then drift detection is likely correct
7. **Evaluation uncertainty:** Synthetic drift scenarios may not perfectly represent real-world market dynamics, limiting the external validity of results.
   - Cross-validate against real historical stress periods (2008 market crisis, COVID crash)
   - Use multiple synthetic drift types (abrupt, gradual, incremental, recurring) to improve generalisability
8. **Using future data (data leakage):** If preprocessing or feature generation accidentally uses future time steps, it can inflate performance metrics and produce misleading conclusions.
   - Separate training, validation, and test sets by time not random sampling

8 Some Idea that may be implemented

1. Two or more swarms can be implemented (e.g., one short-term swarm and one long-term swarm), both sharing the same LSTM backbone. All swarms are trained on the same initial window to ensure consistent feature representations. However, during re-optimisation of the output layer, the short-term swarm uses a smaller adaptation window, while the long-term swarm uses a larger one. This design allows different swarms to capture market behaviour across multiple temporal scales.
2. Consider Ensemble LSTM each trained differently

References

[1] D. Araci and Z. Genç, "Financial sentiment analysis with pre-trained language models," 08 2020.

[2] A. Azeem, I. Ismail, S. S. Mohani, K. U. Danyaro, U. Hussain, S. Shabbir, and R. Z. Bin Jusoh, "Mitigating concept drift challenges in evolving smart grids: An adaptive ensemble lstm for enhanced load forecasting," *Energy Reports*, vol. 13, pp. 1369–1383, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484724008928

[3] M. Bask, L. Forsberg, and A. Östling, "Media sentiment and stock returns," *The Quarterly Review of Economics and Finance*, vol. 93, pp. 302–311, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1062976924000254

[4] M. Davidovic and J. McCleary, "News sentiment and stock market dynamics: A machine learning investigation," *Journal of Risk and Financial Management*, vol. 18, no. 8, p. 412, 2025.

[5] C. Han and B. Fu, "Challenge and opportunity: Deep learning-based stock price prediction by using bi-directional lstm model," *Frontiers in Business, Economics and Management*, vol. 8, pp. 51–54, 04 2023.

[6] S. M. K, R. M. S, A. J. S, and V. K. R, "A novel concept drift detection model for handling evolving patterns in multivariate time series," in *2025 International Conference on Advancements in Power, Communication and Intelligent Systems (APCI)*, 2025, pp. 1–6.

[7] G. Kumar, U. P. Singh, and S. Jain, "An adaptive particle swarm optimization-based hybrid long short-term memory model for stock price time series forecasting," *Soft computing*, vol. 26, no. 22, pp. 12 115–12 135, 2022.

[8] G. H. Oliveira, G. G. Cabral, A. C. Oliveira, M. G. Oliveira, L. L. Minku, and A. L. Oliveira, "Dynamic swarm intelligence for time series forecasting in the presence of concept drift," *SN Computer Science*, vol. 6, no. 6, p. 695, 2025.

[9] N. Rouf, M. Malik, T. Arif, S. Sharma, S. Singh, S. Aich, and H.-C. Kim, "Stock market prediction using machine learning techniques: A decade survey on methodologies, recent developments, and future directions," *Electronics*, vol. 10, p. 2717, 11 2021.